

1. Create following EMP table and perform the following queries .

EMP(Empno, Ename, Job,Mgr,Joindate, Salary, Comm, Deptno)

```
CREATE TABLE EMP ( Empno NUMBER(4),  Ename VARCHAR2(20),  Job VARCHAR2(20),  
Mgr NUMBER(4),  Joindate DATE,  Salary NUMBER(8,2),  Comm NUMBER(8,2),  
Deptno VARCHAR2(5) );
```

```
INSERT INTO EMP VALUES (101, 'Ramesh', 'CLERK', 201, DATE '2022-02-10', 18000, 500, 'D2');
```

```
INSERT INTO EMP VALUES (102, 'Suresh', 'MANAGER', 202, DATE '2021-06-15', 45000, NULL, 'D1');
```

```
INSERT INTO EMP VALUES (103, 'Mahesh', 'CLERK', 201, DATE '2023-02-25', 20000, 800, 'D2');
```

a. Write sql query to list all the job designations in the employee table without repetitions.

```
SELECT DISTINCT Job FROM EMP;
```

b. Write sql query to display all the employees who joined in the month of February.

```
SELECT * FROM EMP WHERE TO_CHAR(Joindate,'MM') = '02';
```

c. Write sql query to sort the employee table in the descending order of salaries.

```
SELECT * FROM EMP ORDER BY Salary DESC;
```

d. Write sql query to display all the clerks in Deptno D2.

```
SELECT * FROM EMP WHERE Job = 'CLERK' AND Deptno = 'D2';
```

2. Create a View on Employee table to display all the employee details along with their work experience in the company till current date.

```
CREATE OR REPLACE VIEW EMP_EXPERIENCE AS
```

```
SELECT Empno, Ename, Job, Joindate,
```

```
TRUNC(MONTHS_BETWEEN(SYSDATE, Joindate)/12) AS Experience_Years FROM EMP;
```

```
SELECT * FROM EMP_EXPERIENCE;
```

3. Write a pl/sql program to find the largest of two numbers.

```
declare
  a int :=10;
  b int :=15;
begin
  if a>b then
    dbms_output.put_line(' a is greater');
  else
    dbms_output_put_line('b is greater');
  end if;
end;
/
```

1. Create following Book table and perform the following queries

BOOK(Book_Id (Primary Key), Book_Name ,Author, Date_Purchased, Publisher ,Price)

For the above table –

```
CREATE TABLE BOOK ( Book_Id NUMBER PRIMARY KEY, Book_Name VARCHAR2(50),  
Author VARCHAR2(50), Date_Purchased DATE, Publisher VARCHAR2(50), Price NUMBER(8,2) );
```

```
INSERT INTO BOOK VALUES (1, 'Database Systems', 'Raghu Sharma', DATE '2023-01-12', 'Himalaya',  
450);
```

```
INSERT INTO BOOK VALUES (2, 'Operating Systems', 'A. K. Sharma', DATE '2022-05-18', 'Pearson',  
550);
```

```
INSERT INTO BOOK VALUES (3, 'Data Structures', 'Sunil Patil', DATE '2023-08-10', 'Himalaya', 6
```

a. Write sql query to display the list of authors from Himalaya publications.

```
SELECT Author FROM BOOK WHERE Publisher = 'Himalaya';
```

b. Write sql query to rename the column Publisher as Publications.

```
ALTER TABLE BOOK RENAME COLUMN Publisher TO Publications;
```

c. Write sql query to create an index on the fields BookName and Author.

```
CREATE INDEX IDX_BOOK_AUTHOR ON BOOK(Book_Name, Author);
```

d. Write sql query to display the book details where author name contains the name Sharma.

```
SELECT * FROM BOOK WHERE Author LIKE '%Sharma%';
```

2. Create a procedure to print the odd numbers from 1 to 10.

```
CREATE OR REPLACE PROCEDURE ODD_NUMBERS AS  
BEGIN  
  FOR i IN 1..10 LOOP  
    IF MOD(i,2) = 1 THEN  
      DBMS_OUTPUT.PUT_LINE(i);  
    END IF;  
  END LOOP;  
END;
```

/

```
EXEC ODD_NUMBERS;
```

3 . Create a view to display the fields BookId and BookName where the Publisher is Himalaya.

```
CREATE OR REPLACE VIEW HIMALAYA_BOOKS AS
```

```
SELECT Book_Id, Book_Name
```

```
FROM BOOK
```

```
WHERE Publications = 'Himalaya';
```

```
SELECT * FROM HIMALAYA_BOOKS;
```

1. Create following Client_Master table and perform the following queries

Client_master (ClientNO, Name, Address, City, State, bal_due)

```
CREATE TABLE Client_master ( ClientNO VARCHAR2(5), Name VARCHAR2(30), Address VARCHAR2(50), City VARCHAR2(20), State VARCHAR2(20), bal_due NUMBER(8,2) );
```

```
INSERT INTO Client_master VALUES ('C101', 'Ajay Shah', 'MG Road', 'Pune', 'MH', 6500);
```

```
INSERT INTO Client_master VALUES ('C123', 'Rina Mehta', 'Station Road', 'Mumbai', 'MH', 4800);
```

```
INSERT INTO Client_master VALUES ('C205', 'Kunal Jain', 'College Road', 'Nashik', 'MH', 7200);
```

a. Find the names of clients whose bal_due > 5000 .

```
SELECT Name FROM Client_master WHERE bal_due > 5000;
```

b. Change the bal_due of ClientNO " C123" to Rs. 5100

```
UPDATE Client_master SET bal_due = 5100 WHERE ClientNO = 'C123';
```

c. Change the name of Client_master to Client12 .

```
ALTER TABLE Client_master RENAME TO Client12;
```

e. Display the bal_due heading as "BALANCE" .

```
SELECT bal_due AS BALANCE FROM Client12;
```

2. Write a pl/sql program to find the largest of two numbers.

(already given , repeated)

3. Create a procedure to print the even numbers from 1 to 20.

```
CREATE OR REPLACE PROCEDURE EVEN_NUMBERS AS
BEGIN
FOR i IN 1..20 LOOP
    IF MOD(i,2) = 0 THEN
        DBMS_OUTPUT.PUT_LINE(i);
    END IF;
END LOOP;
END;
/
SET SERVEROUTPUT ON;
EXEC EVEN_NUMBERS;
```

1. Create following tables and perform queries

(10)

Author(A_id, Author_name, No_of_Books)

Book(Book_id,Book_name,A_id)

```
CREATE TABLE Author ( A_id NUMBER, Author_name VARCHAR2(30), No_of_Books NUMBER );
```

```
CREATE TABLE Book ( Book_id NUMBER, Book_name VARCHAR2(30), A_id NUMBER );
```

```
INSERT INTO Author VALUES (1, 'Prakash Patil', 5);
```

```
INSERT INTO Author VALUES (2, 'Sunita Sharma', 3);
```

```
INSERT INTO Author VALUES (3, 'Amit Kulkarni', 4);
```

```
INSERT INTO Book VALUES (101, 'Java Basics', 1);
```

```
INSERT INTO Book VALUES (102, 'Advanced SQL', 2);
```

```
INSERT INTO Book VALUES (103, 'Python Programming', 1);
```

a. Display the books whose Author's name start with "P".

```
SELECT b.Book_name FROM Book b, Author a
WHERE b.A_id = a.A_id AND a.Author_name LIKE 'P%';
```

b. Display the author's details whose maximum books are published.

```
SELECT * FROM Author WHERE No_of_Books = (SELECT MAX(No_of_Books) FROM
Author);
```

c. Display the books and authors details

```
SELECT a.Author_name, b.Book_name FROM Author a, Book b
WHERE a.A_id = b.A_id ORDER BY a.Author_name;
```

d. sorted in alphabetical order of Author_name.

Use above tables to create View to display the author wise count of books .

```
CREATE OR REPLACE VIEW AUTHOR_BOOK_COUNT AS
```

```
SELECT a.Author_name, COUNT(b.Book_id) AS Total_Books
FROM Author a, Book b
WHERE a.A_id = b.A_id GROUP BY a.Author_name;
```

```
Select * from AUTHOR_BOOK_COUNT ;
```

3. Write a pl/sql program to find the total and average of 6 subjects and display the grade. (10)

```
declare
    s1 int :=65;
    s2 int :=78;
    s3 int :=56;
    s4 int :=68;
    s5 int :=87;
    s6 int :=54;
    avg int;

begin
    avg:= (s1+s2+s3+s4+s5+s6 ) /6;
    if avg >=60 then
        dbms_output.putline ('Grade = A');
    elsif avg <60 AND avg >45 then
        dbms_output.putline ('Grade = B');
    elsif avg <45 60 AND avg >= 35 then
        dbms_output.putline ('Grade = C');
    else
        dbms_output.putline ('Grade = F');
    end if;
end;
/
```

Create table

Teacher (Name, DeptNo, Date of joining, DeptName, Location, Salary)

```
CREATE TABLE Teacher ( Name VARCHAR2(30), DeptNo NUMBER, Date_of_Joining DATE,  
    DeptName VARCHAR2(30), Location VARCHAR2(30), Salary NUMBER(8,2));
```

```
INSERT INTO Teacher VALUES ('Anita Deshmukh', 10, DATE '2019-06-12', 'Mathematics', 'Nashik',  
42000);
```

```
INSERT INTO Teacher VALUES ('Rajesh Patil', 20, DATE '2021-01-10', 'Computer', 'Pune', 50000);
```

```
INSERT INTO Teacher VALUES ('Smita Kulkarni', 10, DATE '2020-09-05', 'Mathematics', 'Mumbai',  
45000);
```

(a) Update the 25% salary for Mathematics Department .

```
UPDATE Teacher SET Salary = Salary * 1.25 WHERE DeptName = 'Mathematics';
```

(b) Display Department wise information of Teachers.

```
SELECT DeptName, COUNT(*) AS Total_Teachers, AVG(Salary) AS Avg_Salary  
FROM Teacher GROUP BY DeptName;
```

(c) Display the teacher's details from Nashik Area.

```
SELECT * FROM Teacher WHERE Location = 'Nashik';
```

(d) Sort the table on Date of Joining and Display the records.

```
SELECT * FROM Teacher ORDER BY Date_of_Joining;
```

2. Write a pl/sql program to find the largest of two numbers.

3. Write a pl/sql program to find the total and average of 6 subjects and display the grade .